NUSC Technical Document 6904
1 June 1990

After-Action Report for the Next-Generation Computer Resources (NGCR) Operating Systems Interface Standard Baseline Selection Process

Operating Systems Standards Working Group (OSSWG)
Compiled by J. T. Oblinger (NUSC)

Naval Underwater Systems Center
Newport, Rhode Island • New London, Connecticut

PREFACE

This report was funded under NUSC Project No. A45146, "Next-Generation Computer Resources (NGCR)." The sponsoring activity is the Space and Naval Warfare Systems Command, through the work of the Operating Systems Standards Working Group (OSSWG). The OSSWG management structure is as follows:

NGCR Program Manager, H. Mendenhall (SPAWAR-324)

NGCR OSSWG Cochairman, CDR R. Barbour (SPAWAR-324)

NGCR OSSWG Cochairman, P. Oberndorf (NADC)

Approach Subgroup Chairman, T. Conrad (NUSC)

Requirements Subgroup Chairman, R. Bergman (NOSC)

Available Technology Subgroup Chairman, J. Oblinger (NUSC)

REVIEWED AND APPROVED: 1 JUNE 1990

PALa Brecque

P. A. La Brecque
Head, Combat Control Systems Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
1 June 1990

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
After-Action Report for the Next-Generation Computer Resources (NGCR) Operating Systems Interface Standard Baseline Selection Process

5. FUNDING NUMBERS
PN A45146

6. AUTHOR(S)
Operating Systems Standards Working Group

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Underwater Systems Center
Newport Laboratory
Newport, RI 02841

8. PERFORMING ORGANIZATION REPORT NUMBER
TD 6904

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Space and Naval Warfare Systems Command
(SPAWAR-324)
Washington, DC 02841

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Next-Generation Computer Resources (NGCR) Operating Systems Standards Working Group (OSSWG) conducted a survey of existing operating systems and operating systems interface standards to establish a baseline for the NGCR operating system interface. This report reviews the OSSWG evaluation process and discusses issues that caused difficulty to OSSWG in meeting its objectives.

14. SUBJECT TERMS
Next-Generation Computer Resources
Operating Systems Interface

15. NUMBER OF PAGES
27

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

EXECUTIVE SUMMARY

The Next-Generation Computer Resources (NGCR) Operating Systems Standards Working Group (OSSWG) conducted a survey of existing operating systems and operating systems interface standards to establish a baseline for the NGCR operating system interface (OSIF). As a result of this survey, the total number of operating systems considered was reduced from 110 to 7, and those final 7 were then formally evaluated. The formal evaluation consisted of assessing the seven candidates against the requirements contained in the "NGCR OSSWG Requirements Document" (reference 1) and a set of eight programmatic issues.

The first section of this report describes the purpose and scope of this study, which covered the timeframe from March 1989 (a briefing made to industry) to April 1990 (when the OSIF baseline was selected).

The second section discusses issues regarding the OSSWG evaluation process. Issues presented include the benefits OSSWG gained by active industry participation, the effectiveness of the electronic mail system for providing communications between meetings, the concerns about the compressed schedule, and a discussion about the difficulty in interpreting the evaluation scores.

The third section addresses the technical issues that caused difficulties for OSSWG in achieving its objectives. Some of these issues include (1) how to define distributed technology within an operating system interface; (2) how to specify security; (3) how security impacts the technology of real-time capabilities, distribution, and fault-tolerance; and (4) to what extent OSIF issues impact the performance of OS implementations. The technology topics in this section are presented as technology shortfall areas where there is need for additional research.

TABLE OF CONTENTS

FOREWORD

The work reported herein was conducted over a period of a little more than 1 year by a joint team of Navy, other government, industry, and academic experts in the field of computer operating systems. Only a few of the Navy participants were actually funded to directly participate in this process. The superb accomplishments of the joint working group and its ability to complete the evaluation process in so short a time span derived from the dedication of all the participants to getting the job done. The outstanding contributions of all of the volunteers in this process are particularly noted and appreciated.

Special thanks are expressed to U.S. industry and academia for their staunch support of and participation in this working group. Their continued support and involvement are strongly solicited.

The OSSWG members who authored sections of this report are:

Charles Arnold	Arnold Assoc.
Richard Bergman	NOSC
Larry Daubert	Rockwell International
LT Karl Fairbanks, Jr.	NWC
Dr. Karen Gordon	IDA
Dr. Mars Gralia	JHU/APL
Daniel Green	NSWC
Joseph Gwinn	Raytheon
Barbara Haleen	Unisys
E. Doug Jensen	Concurrent Computer
Daniel Juttelstad	NUSC
Kari Kruempel	Unisys
Michael Morgan	PICHTR
James Oblinger	NUSC
Frank Prindle	NADC
Carl Reinert	Computer Based Systems
George Robertson	FCDSSA-SD
Helmut Roth	NSWC
Dr. Timothy Saponas	Intel

AFTER-ACTION REPORT
FOR NEXT-GENERATION COMPUTER RESOURCES (NGCR)
OPERATING SYSTEMS INTERFACE STANDARD
BASELINE SELECTION PROCESS

1. INTRODUCTION

1.1 PURPOSE

This report reviews the activities of the Next-Generation Computer Resources (NGCR) Operating Systems Standards Working Group (OSSWG) since its inception in March 1989. The lessons learned that can be passed on to other groups exploring similar territory, as well as the technological shortfall areas where there is a need for additional research, are discussed. In addition, recommendations are made to be used in assisting higher Navy authorities in pursuing solutions in the technological shortfall areas.

1.2 SCOPE

This report is a chronology of significant events from a briefing made to industry in March 1989 to an April 1990 meeting at the Software Engineering Institute (SEI) (Pittsburgh, PA) when the current operating system interface (OSIF) baseline was selected. While candidate evaluation and results analysis are the subjects of other concurrent reports (references 2 and 3), this report focuses on the process of managing the OSSWG and conducting the evaluation. In addition, the activities that significantly enhanced or, in hindsight, hindered progress of the OSSWG are discussed. Recommendations also are provided for research in areas that, if insight had been available, would have assisted the work of the OSSWG.

2. EVALUATION PROCESS ISSUES

This section presents issues felt to be of significance, both positively and negatively, over the current lifespan of the OSSWG. The issues expound on the organization and coordination of the working group and present specifics on the activities leading to the evaluation process and the analysis of evaluation's results. The issues and views presented are composite views of OSSWG participants.

2.1 OSSWG COMPOSITION

The OSSWG is made up of potential end users of systems using NGCR operating systems, people with system development experience, and people with experience in developing operating systems. To put this another way, working group members were drawn from the Navy procurement community; the research community, including representatives from the Navy laboratories, industrial research groups, and the academic community; national standards bodies; and the industrial community responsible for building combat systems for the Navy. Some individuals operated in several of these communities. The definition of the nature and functions of an operating system (OS), the evaluation of areas that were ready for standardization and the selection of potential candidates to serve as a baseline for standardization, require this broad coverage of all the interests involved. That the OSSWG was able to draw on this breadth of participants was seen as a very positive sign.

2.2 EFFECTIVE TEAM

The OSSWG reduced the number of candidate operating system interfaces from more than 100 to 7, crystalized nebulous requirements into 193 specific assessment criteria, and formulated nebulous concerns into dozens of resolved issues. Looking back at that, one can say that the OSSWG has proven to be a very effective, coordinated team. The issues were stated, discussed, and resolved. New approaches were proposed and adopted, adapted, or discarded. Reports were outlined, written, and published.

Obviously, this is not the case with all groups, particularly those with discretionary and unfunded participation. In other standards groups, the transient volunteers often outnumber the permanent members, exacting a heavy penalty as the rookies learn the nomenclature, the scope, and the issues.

One of the lessons relearned by the OSSWG was the value of experienced and mature participants. This fact has many side effects, highlights of which are covered in sections 2.3 and 2.13. This fact is readily documented:

1. The average OSSWG evaluator had 8.8 years of direct work experience with operating systems. Of this, 3.9 years were spent working with an application that intimately used an OS, and 4.9 years were spent in building an OS or its components.

2. Seventy-seven percent of the evaluators work closely with operating systems in their current projects.

It is not clear, however, how this wealth of experience was available at the right time for the OSSWG, or how to make it happen again. But the benefits of such a situation are many, and every working group sponsor should be aware of, and encourage, such participation.

2.3 BIPARTISAN INDUSTRY PARTICIPATION

The complete OSSWG OSIF standard selection process was marked by high quality industrial participation. Of significant note was the bipartisan nature of the participation of those representatives of industry who also served as the point of contact for one of the candidates. This bipartisan participation of the various representatives of industry ensured that a fair and honest examination of technical issues was accomplished. The examination included input from various experts from industry who brought to the process experience with the design and implementation of operating systems. It is believed that the bipartisan nature of the industrial participation was possible because the selection process did not automatically result in the awarding of a specific contract. In fact, it was made clear at the beginning of this process that the awarding of a contract to provide an implementation of the selected standard would be handled independently of the standard selection process.

2.4 ELECTRONIC-MAIL COMMUNICATIONS

The OSSWG used electronic communications (E-mail) for three purposes: as a forum for discussion, for document distribution, and for electronic submission of candidate evaluations. The latter use is perhaps the most noteworthy. Electronic submission allowed evaluators to fill in evaluation forms on their computers, and then E-mail the completed form to the evaluation point. This reduced the cost of compiling the results because no postage was required and no data entry was necessary for evaluations.

The benefits of E-mail as a forum for discussion among geographically dispersed people is well known. The OSSWG found this process an essential method of keeping its membership informed. It would have been impossible to generate discussion on the scale experienced by the OSSWG without this type of communications tool.

The members of the OSSWG who did not have access to the Internet, the national network that allowed E-mail, were given accounts on the NADC mail host. To facilitate mailings, distribution lists were set up. Anyone who wanted to send a message to the entire OSSWG membership could do so by addressing the message to the OSSWG distribution list. The NADC host also was used as a file repository. In addition, many historically significant documents were available for browsing or download. This practice of electronic document distribution reduced cost and increased accessibility.

E-mail traffic may have helped to keep members interested in the program over the duration of the OSIF evaluation. It is believed that the broad distribution of discussions to all the OSSWG members resulted in further discussion that would not have happened with telephone calls or postal mail.

2.5 CONFERENCING SYSTEM

There are some things that the OSSWG may have been done better. The discussions carried out under E-mail were not available after the fact. A historical record of all the traffic posted to the distribution lists would have been of great value to new members and would probably be of general value.

Participation with E-mail is "by message unit" and not "interactive." Because of delays in the mail software, 30 minutes typically would elapse between the time a message was sent and the time when it was received at the last OSSWG member's computer. There is some sentiment that a "party line" style interactive facility might have been more useful. With this technology, many people could communicate with each other immediately and simultaneously.

The E-mail messages were homogenized in the sense that they were not readily differentiated by subject. A corollary was that discussions tended to become defocused. A system should have been set up where each circuit was devoted to a particular topic. Then, so the thought goes, the arguments would always be germane.

One suggestion to reduce these constraints was to use a product like DEC's VAXnotes, which is a "computer-mediated conferencing system" that apparently directly addresses both liabilities while retaining the fault tolerance of E-mail. Another suggestion was to consider "bulletin board" technology. Presumably, there are other products as well that may have improved the communications effectiveness of the OSSWG.

2.6 OSSWG GROUND RULES

The OSSWG ground rules for generating the requirements and subsequent OSIF standard baseline were established at the first working group meeting in March 1989. According to these ground rules, the requirements for the OSIF baseline were to be taken from references 4 through 6. This set of standing ground rules set the scope, applications partitioning, and the OS and applications domains, and gave the direction for the final OSIF product. It also was understood that, in the final analysis, the Navy would have to change the way it handles computer products acquisition and logistics, because weapon system development doctrine will change as the use of interface standards replaces the use of standard computer commodities.

Probably the most difficult part of the requirements development process was to consider only the OSIF and disregard the engineering, implementation, cost, and performance issues by which all of the systems' developers, designers, and operational requirements generators live. Disregarding these issues provided, by far, the most insidious form of "culture shock." Group members and the leadership constantly and firmly reminded each other that the concern was only with the interface. In the end, a constantly repeated phrase was made into a sign for easy reference. It stated "Interface, Not Implementation."

2.7 BOUNDED SCHEDULE AND TASK

The OSSWG has been particularly effective in producing tangible products within a predetermined time schedule. It is believed that this effectiveness has been partially due to the organization and management approach to OSSWG. Navy program management funded a core group of individuals to act as a framework project team to bring work products to the meetings for review. Managing the working group as a project, with the core group managing rather than directing the working group members, worked out well. When the tendency to review past decisions to solve immediate concerns occurred, the OSSWG core encouraged the members to trust the process and the prior decisions and to press on.

Often, interest groups sending representatives to a working group expect that the effort to produce the products of the working group will occur during the working group meeting time. In addition, the progress of the working group often is difficult to manage. It appears that a working group often is more effective as a review team to existing works and opinions, rather than as a creator of work products. The OSSWG overcame these traditional difficulties by taking the project management model and blending it with a facilitative model for handling the meeting activities.

2.8 DEVELOPMENT OF REQUIREMENTS

The initial development philosophy for the generic OSIF requirements was for the requirements subgroup members to start listing any requirement that they considered important to the ultimate development of an OSIF standard. Each requirement submitted had to have a name, definition, rationale, some evaluation or metric information, and bibliography or source information. This approach was an effort to limit the random, transient, or "pet" requirements of any individual in the group. Even with these constraints, the initial number of requirements tallied nearly 1000 (clearly too many to work with). The "percolation" process continued and a grouping called "service classes" was developed. This grouping was used to further refine and control the requirements development process. The service classes that initially encompassed all the requirements numbered as many as 28. As the refinement process continued, the service classes were narrowed down to a final 16 that illustrated the major categories of services, functions, and protocols interface provides or supports for applications and resource management of a system.

The ultimate objective of the requirements subgroup was to develop a reasonable set of service classes. Each of the service classes would support a balanced group of requirements associated with a particular kind of service.

Furthermore, every attempt was made to balance the detail of definition, rationale, and evaluation along with a set of metric parameters of each requirement to minimize the ambiguity of the requirement. Each requirement was to be generic as well as defined for an interface, not an implementation. There also was an attempt made to be aware of the definition of operating systems "terms" and "words," again, to limit the interpretation variations of people with widely ranging experience and background.

The "NGCR OSSWG Requirements Document" (reference 1) was finalized with fewer than 200 requirements, unevenly distributed among 16 service classes. During the evaluation process, the requirements set came fairly close to meeting the aforementioned objectives. Only 12 of the requirements were questioned for wording, definition, or interpretation. The "fixes" to these requirements were relatively minor and will be reflected in an Operational Concepts Document (OCD) to be published in the future.

The requirements development process was made up of a good mix of talented people from government, industry, and academia with the widest range of experience and best cooperation. This was beyond all expectation when compared with similar working groups. The document development milestones were met without sacrificing product quality or completeness.

2.9 REQUIREMENTS FOR DISTRIBUTED AND CENTRALIZED SYSTEMS

Most OSSWG members, because of their experiences, saw the OSIF requirements in terms of the centralized individual uniprocessor and multiprocessor application program interface (API) because of two factors. First, there was a lack of synchronization and coordination of the reference model and the requirements efforts. Secondly, the requirements subgroup made a conscious decision to drop a distribution service class (it was felt that almost all of the requirements would be valid in any portion of a distribution hierarchy).

What did not get done was the development of a graphic to show how the model (as a function of the requirements) and the requirements (as a function of the model) look in any position (application) in the hierarchy of the heterogeneous, open-systems architecture that was specified as the computer resources environment. This was one area that caused significant interpretation problems.

The reference model suffered some "bad press" because the iterative development releases were, in fact, developmental and did not track with the iterative developmental requirements document. The reference model also looked very two-dimensional and local processor operating system (LPOS) oriented because almost no one could visualize how it could be hierarchical. The same is true of the Requirements Document (reference 1).

The requirements were developed with distribution implicit, but it was not obvious to the developers. The suggested graphic(s) will be a part of the OCD. If the requirements had been developed to be distributed, the only differences between LPOS and the system resources allocation executive (SRAX) then will be in the implementation. The system applications have access to the internals of the systems resources; the mission applications do not have this access.

2.10 OSSWG REFERENCE MODEL

At the beginning of the OSSWG process, it was recognized that there was a need for a common way for the group to understand the systems structure, from an OS viewpoint. This would provide a common basis and structure for discussions and deliberations concerning the functions of an OS. Because there was no adequate model available for the group to use for this purpose, so one was generated. This exercise had as its intended result of providing a framework for technical discussions and for understanding where proposed standards fit into the overall scheme of things. This reference model is still under development and is expected to be revised to reflect both a better understanding of what needs to be supplied in the way of standards and changes in the underlying technology. In this sense, the reference model may make some contribution beyond the OSSWG and NGCR.

However, the reference model failed to influence the requirements sufficiently in the sense that, while it identified the role of distribution in the overall OS context, the requirements did not explicitly address distribution. (For more information, see section 2.9.)

2.11 GENERIC SECURITY INTERFACE REQUIREMENTS

This security effort was, without a doubt, the last and most difficult task of the OSSWG. Computer security, by its very nature, is almost totally implementation dependent. All of the accreditation and evaluation criteria and requirements are biased this way.

The security people on the OSSWG were among the best in the security community. These working group members wanted a chance to build security in from the start of the system, not add security on as is done now in developing systems. As a result of the process, they were able to refine almost 1000 of their own requirements from the various "color" documents, which are implementation dependent, into 24 generic interface requirements that can be, for the most part, something other than a binary decision about an interface's effect upon security.

There is still work to be done (especially in the area of security in real-time and distributed systems). However, if there has been any breakthrough-type milestone in the OSSWG effort, it was in the area of security. Assessments about security can now be made before the system, or all of its parts, are fully developed. Test criteria also can be modified or new criteria developed to test the partially completed system for security worthiness as it progresses to completion. The 24 requirements, which are public, will influence future OS design, because of their simple commonsense nature.

2.12 CANDIDATE SCREENING

From the initial OSSWG Brief to Industry in March 1989, the Available Technology (AT) subgroup was tasked with collecting information on available operating system interfaces. A list of existing operating system interfaces was compiled, an Operating Systems for Mission Critical Computing Workshop was held in September 1989 at the University of Maryland, a set of final candidates was determined, and an AT report (reference 8) was generated.

The list of OS interfaces contained 110 operating systems and interface standards. These candidates were selected by members of the AT subgroup and included operating systems from research programs, Navy offices, and commercial vendors. AT subgroup members were assigned specific operating systems and were requested to obtain an interface description based on the August 1989 Requirements Document (reference 7) service classes. Detailed surveys on 21 candidates were generated and were placed in the AT report (reference 8).

The October 1989 AT meeting narrowed the list of operating systems to 10 final candidates (which was later narrowed down to 7 candidates). This process was described as "early screening." To accomplish this early screening, a set of criteria was defined with which to narrow the complete AT list of operating systems interfaces and interface standards to a number manageable for the formal evaluation process. The stated purpose of this process was to narrow the complete list to only those interfaces that have the potential of fulfilling the NGCR OSSWG requirements. Interfaces that were "too close to call" could be expected to be on the candidate list. The formal evaluation would provide the final OSSWG selection.

Several methods were discussed and exercised over several months to be used for narrowing the rather large OSIF candidate list. It was felt that the appropriate method would have a minimum number of criteria, yet provide a fair and equitable separation of the interfaces deemed to be valuable to the OSSWG from those that did not contain solutions for NGCR OSSWG.

The screening as finally accomplished consisted of two separate, though complementary, sets of early screening criteria. The first method was known as the decision option paper (DOP) method and is based on a comparison of OS capabilities against the DOP technology area requirements described in the NGCR DOP document (reference 4). The second method is called the positive negative (PN) method and is based on a set of positive and negative criteria. Positive criteria included topics such as whether the potential candidate is a current interface standard; negative criteria included topics such as if the potential candidate was simply a narrowly focused research tool. Using these criteria, an attempt was made to find the top OSSWG prospects. Both methods arrived at the same set of final candidates.

The candidates selected by this process have proven to be very capable, and have provided valuable insight into their respective philosophies of OSIF design. With the strength of each of the candidates that was selected for in-depth evaluation, there is good confirmation that the early screening process accomplished its intended purpose.

2.13 SCHEDULE

As described in section 2.7, the committee proved to be effective in producing results. But the schedule for the OSIF selection process, from requirements definition through evaluation, was still far too short, particularly given that the process comprised largely volunteer committee activities.

The OSSWG spent a great deal of time defining and documenting requirements for the interface, leaving little time to develop the evaluation criteria. The OSSWG did not delineate service classes as precisely as the evaluation later required nor define all requirement areas with the same granularity. Some requirement areas had many individual criteria and others had few, biasing the areas that contained many criteria. The significance of this was not realized until well into the evaluation; more time upfront would have made this apparent earlier.

The OSSWG adequately limited the available OSIF candidates to a manageable number but, because of time constraints, supporting work that could have increased the confidence in this activity had to be limited. In particular, the working group was unable to provide written surveys for more than a few existing operating systems and operating system interfaces for the AT report (reference 8).

More time was needed to further educate evaluators, define requirements, and establish a consensus on the evaluation criteria. There was only a short time to explain the meanings of the evaluation criteria to the evaluators, too little time to allow for the definition of the candidates to coalesce within clearly defined boundaries, no time for evaluation score consensus building, and little time to allow for mechanical problems such as getting OSIF candidate documentation to all evaluators. Although a short mock-test evaluation was held at the OSSWG meeting in December 1989 and helped correct some procedural problems, a full-length mock-test evaluation would have helped further educate evaluators, establish greater consensus for the evaluation criteria, and correct additional mechanical problems.

2.14 PREDETERMINED PROCESS

To provide as much objectivity as possible and to complete the evaluation process in a predetermined, finite amount of time, it was necessary to generate a predetermined, static evaluation process. This was done with a great deal of care and with as much participation from all OSSWG members as could be obtained. The process met its goals, to a large extent, in the standard's baseline selection. However, the resulting evaluation process had a number of significant limitations and shortfalls. It is felt, however, that significant shortfalls are inherent when one must select the evaluation and decision process in advance in areas as complex as the selection of OSIF standards. While this method of doing business does help significantly to ensure objectivity and timeliness in the conclusions reached, it also means that one is committed to making decisions before all of the information is available. The obvious solution is to allow for a multiple-round process with each round building on the results of the previous round.

In retrospect, the formal evaluation process had the following unexpected outcomes:

1. The answer was less conclusive than expected.

2. The completeness of each candidate in covering all interfaces was emphasized rather than the appropriate coverage of a subset critical for an application domain.

3. The application domains as expressed did not have the expected discriminating power.

4. The averaging process used obscured some features deemed critical.

2.15 CONSIDERATION OF LIMITED FACTORS

The standard selection and evaluation process chosen emphasized the identification of all the operating system interfaces that are required in a large variety of situations and the ability of a single candidate to provide all of these. This process, however, did not consider a number of factors that are critical in an implementation of an OS. Performance, measured by the proper balance of functionality and efficiency, can not be readily evaluated when only the interface definition is known. The evaluation process was not able to address the critical issue as to whether desired functional subsets of the interface standard proposed could be implemented with the required efficiency. (For an additional discussion, see section 3.5.)

2.16 EVALUATION PROCESS

The evaluation process itself (as opposed to its results) played an invaluable role in the OSSWG efforts. The evaluation process was the catalyst that enabled members to learn about and discuss each candidate at some length. There were several notable evaluation process activities.

First, by forcing evaluators to walk through documentation on each candidate and to consider each candidate in terms of many specific requirements, the scoring phase of the evaluation process increased the evaluators' objective knowledge of the candidates. Moreover, and of no less consequence, it helped evaluators to develop stronger subjective or intuitive feelings about the candidates.

Then, in the analysis phase, the evaluation process drove the OSSWG to consider the candidates from another viewpoint. "Why did the representative application domain (RAD) scores turn out so uniform for each candidate?" It was expected that any given candidate would score much higher in some RADs than in others. "What is it about the RADs or about the candidates that makes a candidate's effectiveness indistinguishable with respect to such seemingly divergent RADs?" The struggle with these questions shed new light on the candidates, as well as on the overall task of the OSSWG.

2.17 CREDIBILITY OF RAW SCORES

When confronted with the results of the preliminary analysis, the OSSWG faced the question of "Do the numbers really mean anything?" Three key points caused concern:

1. The dispersion of the raw scores (for a given criterion for a given candidate) was much larger than anticipated or desired. For example, certain candidates had extremely large sigmas, which were often a result of an extremely wide-spread bimodal distribution.

2. In some cases, the raw scores for a candidate clearly failed to accurately reflect its capabilities. For example, the results provided a high mean score for a requirement that clearly should be satisfied by a single interface, but further inspection of the candidate's documentation failed to reveal an interface satisfying the requirement.

3. The RAD scores were uniform across RADs (i.e., for any given candidate, its scores varied little from one RAD to another RAD). See section 3.6 for a discussion of the RAD scores.

It was realized that many factors led to these problems, e.g., imprecisely defined candidates, misleading cross matrixes, unclear requirements, subjective requirements (such as the programmatic issues and many of the general requirements), varying backgrounds of evaluators, and insufficient opportunity to seek an "all in one room" consensus on the evaluation scores.

The following recommendations could have addressed some of these issues:

1. Screening of candidate documentation - Many of the candidates provided a large amount of documentation, much of which had nothing to do with OS interfaces. In some cases, the documentation did not deal with the actual candidate, but rather with systems that ran on top of the candidate. A subcommittee of the OSSWG could have been formed to work with the candidate sponsors to ensure that only proper documentation was passed on to the evaluators.

2. Screening of cross matrixes - All the candidate sponsors eventually provided the evaluators with cross matrixes that attempted to show how the candidate satisfied each of the requirements. The OSSWG recommended that the sponsors provide the cross matrixes but did not specify a format for the document. The result was a set of documents with a wide variety of formats ranging from documents that either explicitly named interfaces satisfying the requirements (or specified explicit pages in the candidate documentation where the interface could be found) to documents that tried to have general discussions about the requirement with suggestions of possible solutions. The latter tended to mislead the evaluators and avoid the issue of clearly identifying the interfaces. This problem could have been addressed by explicitly requiring a cross matrix from each sponsor. This matrix would follow a fixed format that required the candidate sponsor to clearly identify the interface satisfying each requirement.

3. Evaluator education on the requirements - While considerable effort went into the refinement of the requirements and the Requirements Document (reference 1), the evaluators were never formally briefed on all the requirements. It should be noted that at the December 1989 meeting of the OSSWG, the evaluators were provided some clarification on the requirements, but only as part of a forum intended for assigning weights. For this case, the group was divided in half with each subgroup dealing with only half the requirements. This resulted in several different interpretations of the requirements by the evaluators. A briefing would have allowed evaluators to clarify this confusion and moved the group towards a more standard interpretation.

4. Lack of a consensus-building meeting - The evaluators were asked to perform their evaluation in isolation. They were not brought together until after their evaluations were committed. This left no opportunity for the evaluators to share the knowledge they had obtained and adjust their scores. A consensus-building meeting would have allowed this information exchange and moved the OSSWG towards a consensus.

2.18 HIGH STANDARD DEVIATIONS

The high standard deviation (sigma) revealed in the expressed opinions of the evaluators is not necessarily a problem. First, the data in great part were a matter of professional opinion and could not be otherwise when discussing a complex topic that required the reading of masses of source material. The purpose of this evaluation process was to identify the OS efforts that best met current and future military BM/C3 requirements. This identification was accomplished in an acceptable fashion despite high sigmas, as might be expected. Lower sigmas may have been achieved if better documentation and more time were available and, especially, if a process of discussion and consensus building had occurred. It is doubtful, however, whether this would have significantly changed the resultant professional opinion with regard to the existing technology base. This conclusion is probably correct, because the OSSWG was composed of some of the best OS implementors and researchers and no bias was revealed in the data.

2.19 LOGISTICS

The OSSWG evaluation process was severely hampered by logistics problems, contributing heavily to the high sigmas in the scores. It appears that no one appreciated the magnitude of the required logistics effort until very late in the process. Late arrival or nonarrival of candidate OS documentation prevented many evaluators from performing their missions in a timely manner and, by schedule compression, greatly reduced the quality of those evaluations. At the time of the 6-8 March 1990 OSSWG meeting, only half of the evaluations were in, forcing the names of the candidates to be masked and hampering discussions. There were approximately 36 inches of OS documentation to be sent to each of those 71 people who had submitted letters of commitment. This is more than 750,000 pages to be printed and sent. Even with only about one-third of the pages relevant and only one copy of the documentation sent to each site, this still would be approximately 200,000 pages. This activity should have been planned and managed as an explicit exercise in logistics, with deadlines taking into account distribution delays.

The use of the U.S. Postal Service for document distribution, particularly with the size of the packages involved, meant that some OSSWG members did not receive their evaluation package. Many members received empty mailers a month after shipment, or nothing at all. (Most shippers of packages of the size involved use UPS.) Also, the documents should have been shipped in corrugated cardboard boxes, not brown paper, which is too weak to contain such heavy stacks of paper.

3. TECHNICAL ISSUES

This section presents issues for which there is currently insufficient technology to be of assistance to OSSWG. If the technology had been mature, it would have helped the working group to understand and define the OSIF. The areas presented are recommended for funding as research topics, because of their current immaturity.

3.1 DISTRIBUTED/NETWORK SYSTEM MANAGEMENT

Traditional operating systems have managed systems confined to a single processing unit or a well-defined group of processing units. In a distributed or networked system, the OS must handle many of the same kinds of management functions, but with the added dimension of distribution across a varying configuration of processors. A few operating systems have ventured into this area, but more research is needed as the OSSWG discovered when attempting to define the requirements of an SRAX in relation to the requirements of an LPOS. Some of the management functions to be considered for a distributed or networked system include

- System downloading and initialization
- Process control including allowing a single Ada program to be spread across multiple processors
- Workload averaging over processors
- Coscheduling processes across nodes
- Uniform, transparent access to code, data, and devices
- Fault tolerance
- Debugging
- System monitoring and tracing
- Synchronization of execution and data across multiple nodes.

3.2 DISTRIBUTION PARADIGM VERSUS PROBLEM AREA

The NGCR-compliant OS is intended to run in mission-critical Navy platforms, which include a wide variety of computer environments. Platforms of immediate interest include relatively autonomous "fire-and-forget" weapons, such as missiles and torpedoes; relatively unattended platforms, such as satellites; and equipment with many processors and extensive human interfaces, such as the C3I systems of an aircraft carrier. Within this range of platforms and missions, the computers may be simple uniprocessors running high-performance process control loops, closely-coupled multiprocessors, or even widely distributed heterogenous multiprocessors. There may be a single instruction stream (as is the case with uniprocessors and array processors) or each processor may operate on its own instruction stream (as is the case with multiple-instruction stream multiple-data stream (MIMD) machines).

The OSSWG recognized that no single implementation of the OS could handle all applications. An OS suitable for weapons control, including extensive provisions for security and fault tolerance, and running over many processors, would be far too large to run on a uniprocessor in the weapon itself. The OSSWG agreed that an OSIF could be developed as a scalable standard where each application, while conforming to the interface standard, could select the services and functions required for that application.

The OSSIG concept of the range of applications is captured in the "gemstone" series of RADs. One can imagine at least two ways of implementing a family of OS implementations that would run over the RADs. First, one could explicitly build a version of the OS targeted for one or more RADs. Thus, one vendor might specialize in an implementation that was particularly well-suited for RAD "diamond." The same vendor, or another vendor, might build a family member for RAD "ruby." A second possible approach is for a vendor to build a series of OS modules, which are then called by applications code and linked into the load image. The resulting collection of modules would be expected to conform to the interface standard or a scalable selection of the required functions and services. Such an approach would have to deal with conflicts (e.g., two applications assuming different schedulers in the OS) but could lead to a better match between application and system. It is the expectation of the OSSWG that vendors will use this concept of a family as a starting point to develop innovative solutions to Navy and industry needs.

It should be noted that the OSSWG explicitly considered and rejected two interpretations of the family concept. First, the OSSWG did not expect that the components of family members would be compatible between vendors. That is, an applications developer should not assume that one can choose the scheduler module from vendor A and the file system from vendor B. Second, the OSSWG intended that the NGCR OS be primarily aimed at uniprocessors and multiprocessors as these are known today and in the near future. Innovative new architectures such as neural nets are not in widespread use today. While they may perform useful functions in Navy systems someday, it is not expected that the NGCR OS will provide all of the interfaces necessary to build applications on those architectures.

3.3 SECURITY INTEGRAL WITH OPERATING SYSTEM INTERFACES

While progress was made in the OSSWG to evaluate OS security, there remains a tremendous amount of research yet to be accomplished. There are two areas that need to be covered: (1) the security mechanisms that should be designed into an OS when it is being developed, and (2) the feasibility or possibility of designing security into an existing OS design.

The best way to include security interfaces in new systems is to design the system with security as a priority. If that is done, then having a system interface that satisfies security requirements will be a natural outcome. The

National Computer Security Center (NCSC) is evaluating many products where security was incorporated from the products inception. This is not a new research area, but work needs to continue.

Adding security interfaces to existing systems may or may not be difficult based on the security level that the system is to achieve. For example, at B1, one is not as worried about covert channels; therefore, it is much easier to define a security interface for a system that is to achieve B1. At B2 and above, one must worry about covert channels; therefore, it must be ensured that the interface does not produce any of these channels. The time and costs involved may make this impractical, but these efforts are underway. The 1003.6 subgroup and the NCSC effort called Trusix are trying to take an existing interface (in this case, UNIX) and define a security interface for it, but more research is needed to fully understand the limitations.

Another aspect of security that has had limited attention is how security mechanisms work when confronted with other new technologies. How well do the current security solutions work when dealing with fault tolerance, distribution, and real-time techniques, individually or together? In the other direction, what do researchers in real-time distribution and fault tolerance technologies need to know about the requirements imposed by security?

3.4 EFFICIENT OS SUPPORT OF ADA ON MULTIPLE NODES

The OSSWG requirements subgroup recognized quite early that Ada runtime support issues permeated most of the service classes; a decision was made to consolidate these into a separate service class called "Ada Language Support Interfaces." Other service class requirements also were specified with Ada and other high-order language support in mind. It is not clear, however, that the interface requirements, as stated, adequately ensure that an Ada runtime can be built for the NGCR environment to efficiently support all the Ada semantics (including reference 9, chapters 13 and 14).

For example, OS interfaces evaluated that support both "process" and "thread" models of concurrency would appear, at first glance, to more efficiently support the Ada tasking model than interface sets that support only the single (heavyweight) process model. The Ada Language Support Interfaces service class does not aid in making this distinction. On further examination, several of these thread models would not seem to support Ada tasking requirements in multiprocessor or multinode (distributed) systems.

Although it has proven difficult to justify inclusion of efficiency criteria into an interface requirements specification, it is equally difficult to justify a set of requirements that, through oversight, precludes efficient Ada runtime implementations. Indeed, several of the baseline interface standard candidates evaluated completely ignored or sidestepped the Ada runtime interfaces issue. Therefore, further research is needed into the relationship between Ada run-time environments and the NGCR OS interfaces.

An excellent starting point for this is the substantial body of work already done by the Ada Runtime Environment Working Group (ARTEWG) of the Association for Computing Machinery (ACM), and documented in references 10 through 12.

3.5 METHOD TO EVALUATE OS PERFORMANCE

To evaluate the performance of specific implementations of the OSIF that are eventually adopted, a set of performance metrics needs to be created. Emphasis is to be placed on evaluating operating systems at their interfaces where the users, who may be either applications programmers who develop tactical software, or systems programmers who develop system services and system control software above the vendor-provided OS, access the OS services. These user interfaces also can be defined as the calls that may be made on the operating systems' underlying procedures.

In conjunction with the performance matrix, a proof of concept should be investigated that develops a method by which the performance parameters may be determined. Two methods should be developed: one that demonstrates the means by which an OS may be evaluated through simulations, and another through empirical measurement means.

To do this, combat system workloads must be developed that demonstrate realistic circumstances for evaluating OS performance. These workloads should demonstrate pragmatic conditions whereby the OS is expected to work in a tactical situation. The primary difficulty in developing these standard workloads is that the OSIF encompasses the entire range of Navy applications, from smart munitions to large C3 systems. Application domains, possibly the gemstone RADS or some similar variations, will need to be specified to identify the particular subsets of the OSIF for which performance is to be measured.

3.6 REQUIREMENTS APPLICATION DOMAIN MAPPING

The evaluation results showed each OSIF candidate receiving consistent scores across the RAD. The cause of this effort being unable to provide discrimination is rooted in the RAD failing to adequately define individual domains and their requirements. This should have been apparent when weight set 2, comparing each domain to the respective service classes, was developed. Weight set 2 contains values indicating all service classes are of some, or similar, importance to all domains. This is hardly the case. For example, expendable munitions, typically a uniprocessor embedded system, requires only service classes 4, 11, 13, and 14. All others should receive a 0 weight. In fact, if service classes are not capable of being configured out of an OS, then a negative weight may be appropriate. In addition, the criteria for those service classes would differ, if not in function, then in weighting, from the criteria associated with another domain (specifically, a critical criterion for an expendable munition OSIF in its simplicity and corresponding economy of implementation). In contrast, a worldwide communication network-supporting OS would require all 16 service classes, with each service class having its own set of criteria and weights. Economy of implementation, then, is of less value than full functionality.

Domain-specific service class criteria and more discriminating service class weights would have better characterized the support spectrum for each candidate. This approach would have required the specification of the mix of function, performance, fault tolerance, and security for each RAD. In this case, a specific OS technology would have scored consistently across all RADs, only if it were sufficiently evolvable, flexible, and extensible to meet a multiplicity of requirements. Thus, the degree to which the OS concept supported evolvability would have been revealed.

3.7 AREAS OF INDUSTRY AND MILITARY OVERLAP

To a large extent, commercial technologies do not have the same need that have driven military systems in many years, i.e., the need for real-time/critical-time processing, where hard deadlines must be met or mission failure and loss of life are the outcome. Some overlap of commercial systems is being recognized in areas such as nuclear reactor controls, manufacturing control systems, and banking systems. Real-time processing is becoming more important in these areas for several reasons: reactor control mechanisms must constantly monitor the reactor to prevent an accident; automated manufacturing requires parts to come together at a specific time to maintain a smooth flow of quality products to remain profitable; and banking customers do not want to spend all day waiting for their withdrawals from an automated teller machine. This added commercial demand is creating a stronger market for the kind of systems the military has needed.

Military systems are not only concerned with real-time capabilities but other features as well, such as fault tolerance, security, distributed systems, Ada, heterogeneity, and large real-time data bases in the gigabyte range. While commercial operations such as banking and nuclear reactor control systems have some similar requirements, in general, only a subset of the military requirements are necessary for commercial applications. Commercial applications might be concerned with one or two requirements, such as fault tolerance, distribution, or soft real-time processing, whereas, military systems frequently require all those capabilities constantly. The interaction of these sometimes-conflicting requirements (e.g., real-time performance, distribution, and security) can have an adverse impact on performance; the additional requirement of Ada only complicates the problem. A large portion of industry standardization efforts has been influenced by UNIX, and it is often tightly coupled to the 'C' programming language. The resulting standards are usually expressed in 'C' bindings, not in Ada.

To meet all the requirements, further research should be conducted to determine how the various needs of a military system interact. The interfaces must be defined in such a way that an appropriate mix of these requirements can be specified for each application domain.

A major issue is that military (combat) systems are subject to overt hostile attack and civilian ones are not. (Also, combat is a far more dynamic and stochastic environment than any civilian one in the sense that its mission is so subject to change.) Despite this, military combat systems demand the ultimate in dependable effectiveness, safety, and survivability -- no civilian applications do more than approximate this need (and they are not under overt hostile attack).

REFERENCES

1. "NGCR OSSWG Requirements Document," SPAWAR-324 Informal Document, Version 2.0, 21 December 1989.

2. "Evaluation Process Report for NGCR OSIF Baseline Selection," SPAWAR-324 Informal Document, Version 1.0, 7 May 1990.

3. "Evaluation Results Report for NGCR OSIF Baseline Selection," SPAWAR-324 Informal Document, Version 1.0, 7 May 1990.

4. "Tentative Operational Requirements for Next-Generation Computer," SPAWAR ltr to NCGR OSSWG, Ser 324.253, 30 October 1987.

5. "NGCR Development Options Paper," CNO ltr to SPAWAR, Ser 098r5u35846, 31 December 1985.

6. "White Paper on Network Operating Systems Standards," Naval Ocean Systems Center, San Diego, CA, August 1988.

7. "NGCR OSSWG Requirements Document," SPAWAR Draft Document, August 1989.

8. NGCR Available Technology Report," SPAWAR-324 Informal Document, Version 1.1, 18 April 1990.

9. "Ada Reference Manual," ANSI/MIL-STD-1815A-1983, American National Standards Institute, New York, 1983.

10. "A Model Runtime System Interface for Ada," Ada Runtime Environment Working Group, MRTSI Taskforce, Informal Report, Version 2.3, Association for Computing Machinery Special Interest Group for Ada, 11 October 1988.

11. "A Catalog of Interface Features and Options for the Ada Runtime Environment," Ada Runtime Environment Working Group, MRTSI Taskforce, Release 2.0, Association for Computing Machinery Special Interest Group for Ada, Decembr 1987.

12. "Catalogue of Ada Runtime Implementation Dependencies," U.S. Army Headquarters. Center for Software Engineering, Advanced Software Technology, Fort Monmouth, NJ, 15 October 1988.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
CNO (NOP-00, NOP-112, NOP-22, NOP-211, NOP-224, NOP-35, NOP-351, NOP-04, NOP-55, NOP-551, NOP-094, NOP-940, NOP-941, NOP-942, NOP-944, NOP-945, NOP-095, NOP-098, NOP-982, NOP-983)	20
CNR (OCNR-1133 (A. Van Tilborg))	2
U.S. Marine Corps HQ (C21, MCRDAC, PSE, LMA-1, CCA-51, OCIER, MCTSSA)	7
DIRSSP (NSP-00)	1
Cruise Missiles Project (Attn: C. Thomas, B. Tung)	2
NRL (Code 5540B (H. Lubbes), Code 5544 (M. Voreh))	2
COMNAVSECGRU	1
DONIRM (Attn: T. Senator)	1
PRESINSURV	1
INSURVLANT (Attn: Senior Member)	1
INSURVPAC (Attn: Senior Member)	1
NTISA, San Diego	1
COMNAVTELCOM	1
COMNAVAIRSYSCOM (NAIR-00 (30 copies), NAIR-5466 (B. Corson), NAIR-54664 (V. Skullman), PMA-209E (B. Anderson, M. Emerson))	34
COMSPAWARSYSCOM (SPAWAR-003-22, PD-40, PMW-141, PMW-142, PMW-143, PMW-144, PMW-145, PMW-146, PMW-147, PD-50, PMW-151, PMW-152, PMW-153, PMW-155, PMW-156, PMW-159, PD-60, PMW-161, PMW-162, PMW-163, PMW-164, PMW-610, PMW-620, PMW-630, PD-70, PD-70C, PD-70-1, PD-70-2, PD-70-3, PD-70-4, PMW-174, PD-80, PMW-180, PMW-181, PMW-182, PMW-183, PMW-184, PMW-162-13A (F. Deckelman), PMW-183-421 (R. Cockerill), SPAWAR-324 (P. Andrews, N. Stopyra), SPAWAR-324A (R. Barbour), SPAWAR-32431B (K. Chan), SPAWAR-3243 (A. Lewin, H. Mendenhall), SPAWAR-3243B (R. Owens, P. Pell), SPAWAR-3212 (M. Romeo), SPAWAR-32 (G. Wagner))	49
COMNAVSUPSYSCOM	1
COMNAVSEASYSCOM (SEA-00 (30 copies), SEA-90, SEA-06D1, PMS-412)	33
DTRC, Bethesda	3
DTRC, Annapolis	3
NADC (Code 7031 (P. Oberdorf), Code 7032 (F. Prindle), Code 7033 (C. Schmiedekamp), Library (3))	6
NCSC (Library (3))	3
NOSC (Code 424 (R. Bergman), Code 855 (W. Fitzgerald, R. Hall), Code 41 (A. Justice), Code 412 (W. Loper, D. Wilcox), Library (3))	9
NSWC, Dahlgren (Code N305 (D. Green), Library (3))	4
NSWC, White Oak (Code U33 (S. Howell, H. Roth), Library (3))	5
NAVWPNCEN (Code 3922 (K. Fairbanks), Code 3922 (C. Hall), Code 31C (J. Zenor), Library (3))	6
NSWSES	1
FCDSAA, San Diego (Code 8D (G. Robertson))	1
FCDSAA, Dam Neck	1
INTCOMBATSYSTESFAC	1
NAVELEXCEN, Charleston	1
NAVELEXCEN, Portsmouth	1

Addressee	No. of Copies
NAVELEXCEN, San Diego	1
NAVELEXCEN, Vallejo	1
NAVELEXCEN, Mayport	1
NAVELEXCEN, Pawtuxet River	1
NAVELEXCEN, Mechanicsburg	1
NAVELEXCENACT, St. Inigoes	1
NAVELEXCENACT, Philadelphia	1
NAVELEXSECCEN	1
NAVMASSO	1
NAVSPASYSACT	1
NATC (Code SY30 (J. Abrams, L. Campbell))	2
PMTC (Code 4003 (L. Cook))	1
NAVAVIONICCEN (Code 825 (J. Johnson))	1
COMNAVINTCOM	1
CNET	1
NETC	1
NETSAFA, Pensacola	1
NPS (Computer Science Dept, Administrative Science Dept, Engineering Dept, Library)	4
COMOPTEVFOR, Norfolk	1
ASSTSECNAV (S&L)	2
ASSTSECNAV (R,E,&S)	2
DSMC	1
APL, Johns Hopkins (Attn: B. Shaw, E. Conn, M. Gralia, G. Masson)	4
Commandant, U.S. Coast Guard (Code-G-FQA, G-TES (R. Buddeberg))	2
COMNAVDAC (Code 14 (P. Robinson))	2
NARDAC, Norfolk	1
NARDAC, Pensacola	1
NARDAC, San Diego	1
NARDAC, Alameda	1
NARDAC, Jacksonville	1
NARDAC, New Orleans	1
NARDAC, Pearl Harbor	1
NARDAC, Newport	1
NAVAIRENGCEN	1
NOS, Indian Head (Code 524)	1
NOS, Louisville (Code 50 (J. Shea))	1
DARPA (W. Scherlis)	1
USAF (AFSC/ENR) (Attn: P. Vaccaro)	1
USAF (SSC/XPT) (Attn: E. Crouse)	1
USAF (SAF/AQXA) (Attn: R. Gross)	1
Rome Air Development Center (COTD) (Attn: T. Lawrence)	1
USAISC (Attn: T. Fong)	1
NASA (Langley Research Center) (Attn: S. Beskenis)	1
Booz, Allen, & Hamilton (Attn: S. Alba, M. Karan, E. Rodriquez))	3
Institute for Defense Analysis (Attn: K. Gorden)	1
Arnold Associates (Attn: C. Arnold)	1
TRW Systems Division (Attn: S. Barry, A. Marmor-Squires)	2
Rockwell International (Attn: S. Bishop, L. Daubert, C. Dodd, F. Martin)	4

Addressee	No. of Copies
Carnegie-Mellon University (Attn: M. Borger, H. Tokuda)	2
Vitro (Attn: C. Brown, T. Darby, W. Greenspan, D. Sassaman)	4
CAE-Link Flight Simulation (Attn: L. Burns)	1
SOFTECH (Attn: B. Buseman)	1
MITRE (A. Carangelo)	1
IBM Manassas (Attn: D. Carrow, P. Watson)	2
ESL (Attn: G. Caswell)	1
UCLA (Attn: W. Chu)	1
DGM&S (Attn: S. Davis)	1
Telesoft (Attn: K. Dixon, R. Hadley)	2
G.E. Corporate Research & Development (Attn: W. Dixon)	1
Clemson University (Attn: T. Drake, J. Leathrum)	2
CEA (Attn: G. Driskill)	1
Intel (Attn: R. Dvorchak, T. Saponas)	2
Open Software Foundation (Attn: W. Finley)	1
Honeywell Federal System (Attn: L. Fraim)	1
DY-4 Systems (Attn: K. Clohessy, J. James)	2
Compumetrics (Attn: G. Coraluppi)	1
American Systems Corp. (Attn: M. Cornelison)	1
Martin Marietta Energy Systems (Attn: M. Cossey)	1
Control Data (Attn: R. Cunius, G. Jaeger)	2
Concurrent Computer (Attn: B. Dasarathy, E. Jensen)	2
Raytheon (Attn: I. Davis, J. Gwinn, L. Rainville)	3
Cray Research (Attn: C. Garrett)	1
Ready Systems (Attn: S. Gascon, J. Ready, D. Vrsalovic)	3
Computer Sciences (Attn: B. Gladstone, W. Lev, P. Wood)	3
Dynamics Research (Attn: R. Gretlein, G. Lampshire, P. Palatt)	3
Unisys (Attn: B. Haleen, M. Kamrad, K. Kruempel, J. Lonjers, H. Polzer, D. Swanson)	6
NIST Software Engineering (Attn: J. Hall, G. Fisher)	2
Integrated Software (Attn: S. Harbaugh)	1
Litton Data Systems (Attn: N. Henderson, A. Ruemke)	2
G.E. Ocean Systems Division (Attn: H. Hollander)	1
Texas Instruments (Attn: J. Jensen, K. Johnson)	2
UC Irvine (Attn: K. Kim)	1
Honeywell (Attn: D. Knudson)	1
Oracle (B. Kowalski, D. Reilly)	2
University of Pennsylvania (Attn: I. Lee, N. Prywes)	2
Electronic Data Systems (Attn: N. Lesher)	1
IBM (Attn: D. Locke)	1
Jet Propulsion Laboratory (Attn: F. Mathur, J. Rohr)	2
Advanced Technology & Research (Attn: R. Mayhall, A. Meskin)	2
SCTC Inc. (Attn: B. Miracle)	1
PICHTR (Attn: M. Morgan)	1
G.E. Advanced Technology Laboratories.(Attn: J. Nixon)	1
Numerix Federal Systems (Attn: D. Olson)	1
SYSCON (Attn: R. Owens)	1
Digital Equipment (Attn: J. Reed)	1
Computer-Based Systems (Attn: C. Reinert)	1
Hewlett-Packard (Attn: L. Reveron)	1

Addressee	No. of Copies
Teledyne Brown Engineering (Attn: E. Rozelle)	1
BBN Systems & Technologies (Attn: K. Schroder, S. Vinter)	2
Mnenomics (Attn: R. Semmes)	1
Industrial Programming (Attn: C. Sigda)	1
Logicon (Attn: B. Stauffer)	1
Trusted Information Systems (H. Tajalli)	1
Data General (Attn: L. Trumbore)	1
Motorola (Attn: R. Vanderlin)	1
IBM-FSD (Attn: D. Vogel)	1
Odyssey Research Associates (Attn: D. Weber)	1
Systems Exploration (Attn: B. Welty)	1
Westinghouse Electric (Attn: T. Wishart)	1
Advanced Systems Technologies (Attn: G. Wright)	1